

Pulling Levers for Coins

Problem ID: a13p01pullingleversforcoins

Add the following functionality to tiles $(1, 2)$, $(2, 2)$, $(2, 3)$ and $(3, 2)$:

When the player enters these tiles, offer them the choice to pull a lever. If they choose to pull the lever, they are to be told that they received one coin.

There is no limit on how often the player can get a coin from each lever. But they should only be presented with the option to pull a lever upon entering the room. So if they want to be asked again, they must first go away, to another tile, and come back. Picking an invalid direction (walking into a wall) does not count as leaving the room.

In addition, you need to keep track of the number of coins the player has and display that, each time a lever is pulled. When the goal tile has been reached, the program also prints out how many coins the player retrieved.

Input

The input consists of a sequence $l = (l_1, \dots, l_n)$ of n lines, where each line l_i should be interpreted as containing either an answer to the question where the player wants to move or an answer to the question whether the player wishes to pull a lever, depending on the state of the game at the time when that line is encountered.

In the tests, the last line of input will always result in the player reaching the goal tile. It is guaranteed that the player will reach the goal tile in at most 200 moves.

Output

The quick way to see how the output should be, is to have a look at the given samples. But if you run into any problems, the following elaboration should help clarify.

More detailed output description

The output consists of m lines, which form a pattern consisting of the following building blocks:

- A direction block D , consisting of 2 to 3 lines:
 - A direction menu M_D .
 - A direction prompt P_D .
 - An invalid direction alert A_I (conditional).
- A lever block L , consisting of 1 to 2 lines:
 - A lever prompt P_L .
 - An income report R_I (conditional).
- A victory report R_V .

A more detailed description of these blocks is given below, but first we will describe the overall pattern.

- The first block should be a direction block.
- The last block (and only the last block) should be a victory report.
- In between should be a sequence of direction blocks sometimes interleaved with a lever block.
- There should never be two lever blocks in a row, without a direction block between them.
- Any direction block containing an invalid direction alert should be immediately followed by another direction block.
- A lever block should never be preceded by a direction block containing an invalid direction alert.
- However, the absence of an invalid direction alert does not automatically mean that a lever block should follow.

- A lever block should only be presented when the corresponding game state, as kept track of by the program, indicates that the player is located on a tile with a lever.

So, the output should form a sequence $b = (b_1, \dots, b_k)$ of k blocks, where for each i with $1 \leq i < k$, the following rules should be observed:

- If b_i is a lever block or a direction block containing an invalid direction alert, then b_{i+1} should be a direction block.
- If b_{i+1} is a lever block or victory report, then b_i should be a direction block without an invalid direction alert.

Here is a more detailed description of these blocks:

- A direction block D always begins with a direction menu, followed by a direction prompt.
 - A direction menu M_D consists of the line “You can travel: {v}.”, without quotations, where v is a list of the directions available from the current tile, separated by spaces and the word “or”. In the presentation, the valid directions should be capitalized with the first letter inside parentheses: (N)orth, (E)ast, (S)outh, and (W)est.
For example, if the available directions are north, south and west, the direction menu consists of the line “You can travel: (N)orth or (S)outh or (W)est.”,
 - A direction prompt P_D just consists of the line “Direction: ”, without quotations. The space at the end is optional, but there needs to be at least some whitespace to separate this line from the next line of the output. Of course, as long as the next “line” of the output does not appear on the same line, it is already separated by a carriage return (new line character).

Then, depending on the corresponding answer from the input, line l_i for block b_i , an invalid direction alert should follow if the input l_i does not correspond to a valid direction selection. The comparison should not be case-sensitive. The total set of valid choices is N, E, S, and W, as well as their lower-case counterparts, but depending on the location of the player, the options will be restricted to various subsets of these choices.

- An invalid direction alert A_I consists of the line “Not a valid direction!”
- A lever block L always begins with a lever prompt.
 - A lever prompt P_L just consists of the line “Pull a lever (y/n): ”, without quotations. The space at the end is optional, as with the direction prompt.

Then, depending on the corresponding answer from the input, line l_i for block b_i , an income report should follow if the input l_i is “y”, where the comparison should not be case-sensitive.

- An income report R_I consists of the line “You received {c}, your total is now {t}.”, where c is the amount of coins received from the lever, and t is the total amount of coins the player has received so far, since starting out at the starting location.
- A victory report R_V consists of the line “Victory! Total coins {T}.”, without quotations, where T is the total amount of coins the player received throughout their journey from the starting location to the goal tile.

As should be clear from the above, k should be equal to $n + 1$, that is, the number of blocks that form the output should be one more than the lines of input, as each block will correspond to (and “consume”) one line of input, except the final block, the victory report.

Sample Input 1

s
n
Y
N
w
E
n
e
s
Y
S

Sample Output 1

You can travel: (N)orth.
Direction:
Not a valid direction!
You can travel: (N)orth.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 1.
You can travel: (N)orth or (E)ast or (S)outh.
Direction:
You can travel: (E)ast or (S)outh.
Direction:
Not a valid direction!
You can travel: (E)ast or (S)outh.
Direction:
Pull a lever (y/n):
You can travel: (E)ast or (W)est.
Direction:
You can travel: (S)outh or (W)est.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 2.
You can travel: (N)orth or (S)outh.
Direction:
Victory! Total coins 2.

Sample Input 2

n
Y
e
Y
w
n
n
e
Y
e
s
n
s

Sample Output 2

You can travel: (N)orth.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 1.
You can travel: (N)orth or (E)ast or (S)outh.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 2.
You can travel: (S)outh or (W)est.
Direction:
Pull a lever (y/n):
You can travel: (N)orth or (E)ast or (S)outh.
Direction:
You can travel: (E)ast or (S)outh.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 3.
You can travel: (E)ast or (W)est.
Direction:
You can travel: (S)outh or (W)est.
Direction:
Pull a lever (y/n):
You can travel: (N)orth or (S)outh.
Direction:
Victory! Total coins 3.

Sample Input 3

n
y
e
y
w
n
n
e
y
n
e
s
n
w
s

Sample Output 3

You can travel: (N)orth.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 1.
You can travel: (N)orth or (E)ast or (S)outh.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 2.
You can travel: (S)outh or (W)est.
Direction:
Pull a lever (y/n):
You can travel: (N)orth or (E)ast or (S)outh.
Direction:
You can travel: (E)ast or (S)outh.
Direction:
Pull a lever (y/n):
You received 1 coin, your total is now 3.
You can travel: (E)ast or (W)est.
Direction:
Not a valid direction!
You can travel: (E)ast or (W)est.
Direction:
You can travel: (S)outh or (W)est.
Direction:
Pull a lever (y/n):
You can travel: (N)orth or (S)outh.
Direction:
Not a valid direction!
You can travel: (N)orth or (S)outh.
Direction:
Victory! Total coins 3.